Modified Blowfish algorithm analysis using derivation cases

Theda Flare G. Quilala, Rogel L. Quilala
College of Computer Studies, Tarlac State University, Philippines

Article Info

Article history:

Received Feb 20, 2020
Revised Apr 29, 2021
Accepted Jun 14, 2021

Keywords:

Avalanche
Decryption
Encryption
S-box
Security

ABSTRACT

This study analyzed and enhanced the modified Blowfish algorithm (MBA) encryption. The modification retained the original structure, process and the use of two S-boxes in the MBA but presented two derivation processes in the f-function which was originally placed to prevent symmetry. The derivation case's performance was analyzed using avalanche effect and time efficiency. After comparing the first and second derivation process presented in the MBA, the second derivation further improved the avalanche effect by 5.47%, thus improving security. The performance also showed that the second modification is faster by 39.48% in encryption time, and 38.34% faster in decryption time. The first derivation case in the modified Blowfish was slower in time because of the difference in the placement of the shift rotation. The key generation time was found to be independent of the input size while the encryption and decryption time was found to be directly proportional to file size. With this, the second modification is considered to be better.

Corresponding Author:

Theda Flare G. Quilala
College of Computer Studies
Tarlac State University
Romulo Blvd., San Vicente, Tarlac City 2300, Philippines
Email: tfgquilala@tsu.edu.ph

1. INTRODUCTION

Digital communications use has escalated over the years, and this has put more attention on security issues [1], [2]. Concerns such as stealing of personal information, bank account details, and even identity theft surfaced, but these were addressed by providing security upon digital communication channels [3]. Application and use of cryptography in securing information during transmission protects data against known attacks and reduces the risk of hacking [4]-[6].

Cryptography presents a means of protecting sensitive information by transformation, making text unintelligible using certain mathematical algorithmic processes, and the appropriate key to transform again into readable text [7]-[8]. Application of cryptography ensures confidentiality, data privacy and secure information exchange [9]-[15]. Cryptographic techniques involving symmetric and asymmetric encryption, ensures the privacy of data [16], [17]. Among symmetric encryption, popularly used cryptographic algorithms are DES, 3DES, AES, RSA, and Blowfish, each has weakness and strength [18]. Between these, experimental results and comparison proved Blowfish algorithm the best considering time [19].

Designed by Bruce Schneier, the Blowfish algorithm was originally created to replace the outdated DES in 1994. Blowfish is characterized by the use of 64-bit variable-length symmetric key block cipher [20]. Blowfish is easy, simple and fast and consequently, a free alternative to existing encryption algorithms that feature variable security levels, except when changing keys [21]. Numerous researches conducted performance comparisons based on different evaluation parameters to test the security aspect and speed provided by Blowfish, and results showed it is undeniably fast and secure [7], [22], [23].

Even though Blowfish is considered a remarkably fast block cipher, the current standard requires a minimum of 128-bit block size [24], [25] which renders Blowfish unsuitable because it can only accommodate 64-bit block, a quality seen undesirable [26] because it may lead to duplicate blocks that will eventually make other forms of attacks possible [27] consequently compromising data security. Although Twofish, an algorithm related to Blowfish, accepts 128-bit block size and provides a good level of security, it nonetheless lacks encryption speed as compared to Blowfish [28]. Several researchers have attempted to extend the block size of Blowfish to 128-bit [25], [29], [30] results indicate a considerable increase in time and need for larger memory, which makes the performance more unfavorable for use in application that prioritizes speed and makes it inefficient for use in small devices with a little memory.

One study has modified the Blowfish algorithm [31], [32] by using two S-boxes and adding a derivation technique. This study used a 128-bit block size and addresses speed and memory use. However, this study can be enhanced further by analyzing different derivation techniques. Specifically, this paper sought to determine the improvement between two derivations in terms of avalanche effect and time efficiency. The use of a 128-bit block size can help encrypt files larger than 32 GB to lessen the probability of having duplicate blocks, thus improving security. The study is beneficial to organizations when the modified Blowfish algorithm (MBA) with derivation is used as a cryptographic algorithm to secure the information saved on their servers, since cryptography addresses issues of data privacy preservation and encryption of records for transmission over the public network infrastructure.

2. RESEARCH METHOD

2.1. Materials

Software implementation of the modification was carried out using Visual Basic 6.0. The operating system used was running a 64-bit Windows 7 on a PC with an AMD A10-7860 K Radeon R7, 12 Compute Cores 4C+8 G 3.6 GHz processor with 8.00 Gb RAM. For the avalanche test, three sets of plaintext messages against five keys were used, varying 1 bit for each key. In testing the time efficiency, different text files with sizes ranging from 10 kB to 1000 kB were utilized to test the speed of the derivation algorithms.

2.2. Research procedure

2.2.1. Design of modified Blowfish algorithm encryption and decryption

The distinction with the original algorithm is the size of the input block. From the then 64-bit, the input block was incremented to 128-bit and then divided into two equal 64-bit segments, left (LE0), and right (RE0). After that, LE0 was XORed to P1 and P11 in the P-array, all entries in the P-array consists of 32-bit entries. Then, the 64-bit result of the XOR operation with P1 and P11 was inputted to the F-function. Next, the output from the F-function was XORed with the RE0 of the input block. Following this is the swapping of LE0 and RE0. The process was repeated eight times. After the eighth round, LE8 and RE8 were swapped to reverse the last swap. Then, RE8 was XORed to P9 and P19 of the P-array, and LE8 was XORed to P10 and P20. Finally, LE9 and RE9 were joined to produce the 128-bit ciphertext. The decryption process follows the inverse of the encryption process. Figure 1 showed the MBA encryption and decryption procedure.

2.2.2. F-function

Figure 2 and Figure 3 showed the details in the construction of the new F-function in the modified Blowfish. The figure also showed the difference between derivation process 1 and process 2. Derivation of the S-boxes was done at runtime by a simple rotation. Rotations were in the input or output, either left or right, by one position.

Two modifications were presented. For both modifications, it is clearly seen that the F-function now takes a 64-bit data stream as input and was later subdivided into eight 8-bits (*a, b, c, d, e, f, g,* and *h*). In both figures, *a* was assigned as the first 8 bits, *b* was the next 8 bits, up to the last 8 bits. As each 8-bit data bits were entered into the S-box, it was transformed into a 32-bit data value. The first half (*a, b, c,* and *d*) used the first S-box, while the next half (*e, f, g,* and *h*) utilized the second S-box. For the derivations, some variables are shifted to the left or right before inputted to the S-box. Other variables are also shifted either to the left or right after the S-box. The 32-bit value produced by the S-box 1 for *a* was then XORed, added, and XORed to the output of *b, c,* and *d*, after subjecting their values to S-box 1. This process produced the final 32-bit value for S-box 1. The same procedure was done for S-box 2, but used values for *e, f, g,* and *h*. The final 64-bit output was the concatenated values from S-box 1 and S-box 2. The structure of the F-function has changed, as reflected in (1) and (2).

$$\begin{aligned}&\mathrm{IF(LE0)}=((\mathrm{S1}(a)+\mathrm{S1}(b)\ll 1 \bmod 2\hat{\ }32\ )\ \mathrm{XOR}\ \mathrm{S1}(c)\gg 1)+\mathrm{S1}(d\ll 1) \bmod 2\hat{\ }32\ |\\&((\mathrm{S2}(e)+\mathrm{S2}(f)\ll 1 \bmod 2\hat{\ }32\ )\ \mathrm{XOR}\ \mathrm{S2}(g)\gg 1)+\mathrm{S2}(h\ll 1) \bmod 2\hat{\ }32\end{aligned} \tag{1}$$

$$F(LE0)=((S1(a)+S1(b)\ll 1 \bmod 2\hat{}32)\ \mathrm{XOR}\ S1(c \ll 1))+S1(d\gg 1) \bmod 2\hat{}32 \mid ((S2(e)+S2(f)\ll 1 \bmod 2\hat{}32)\ \mathrm{XOR}\ S2(g \ll 1))+S2(h\gg 1) \bmod 2\hat{}32 \quad (2)$$

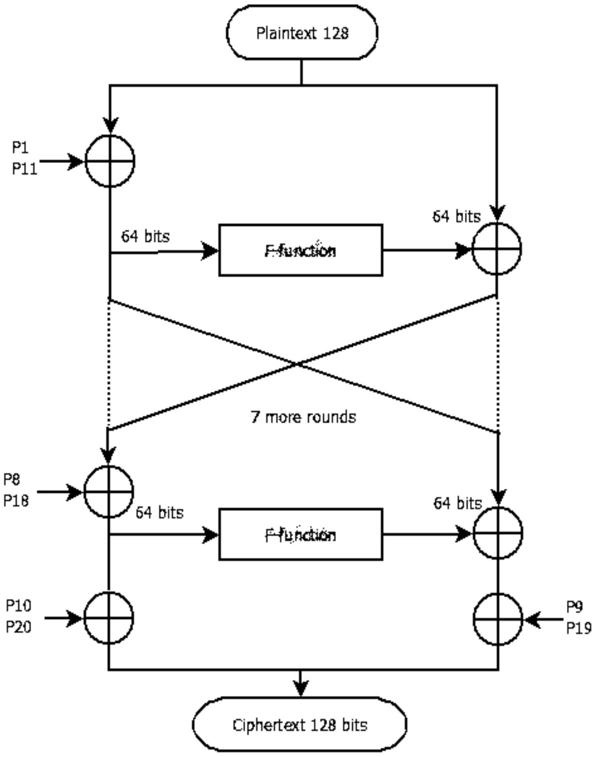


Figure 1. Blowfish modification using 128-bit block size

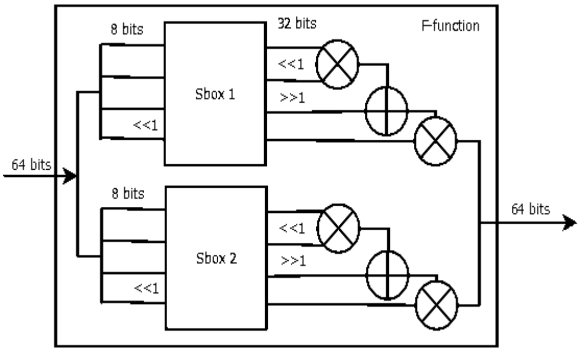


Figure 2. Blowfish F-function using 128-bit block size with derivation process 1

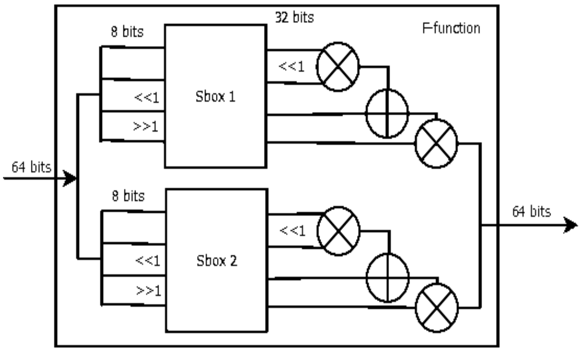


Figure 3. Blowfish F-function using 128-bit block size with derivation process 2

2.2.3. Key expansion

The key expansion process in the modified Blowfish converted the 128-bit key length into several subkey arrays. The modification was able to lessen the number of bytes used from the previous 4168 bytes down to 2128 bytes. The modification only used 20 values in the P-array (P1, P2…P20), each entry consists of 32-bit subkeys and two S-Boxes, each also consisted of 256 entries (S1 - 0…255, S2 - 0…255) of 32-bits each. In the new expansion scheme, using the modifications, the number of iterations to generate all required subkeys was reduced from 521 down to 266. This signifies less storage requirement for the P-array and S-boxes. Calculation of the subkeys was done using the same Blowfish algorithm, using the two S-boxes and the two variants of the derivation process.

2.3. Research design

Figure 4 shows how the study was designed and conducted. As can be seen in Figure 4, the first step includes the enhancement of the MBA by creating two different derivation cases. After the application of the enhancement in the MBA, the next step is the evaluation of these test cases using an encryption program. Avalanche effect and speed are the performance parameter used using text string and different encryption keys, and text files of different sizes. All of this will be done to produce an enhanced MBA with improved speed and security.

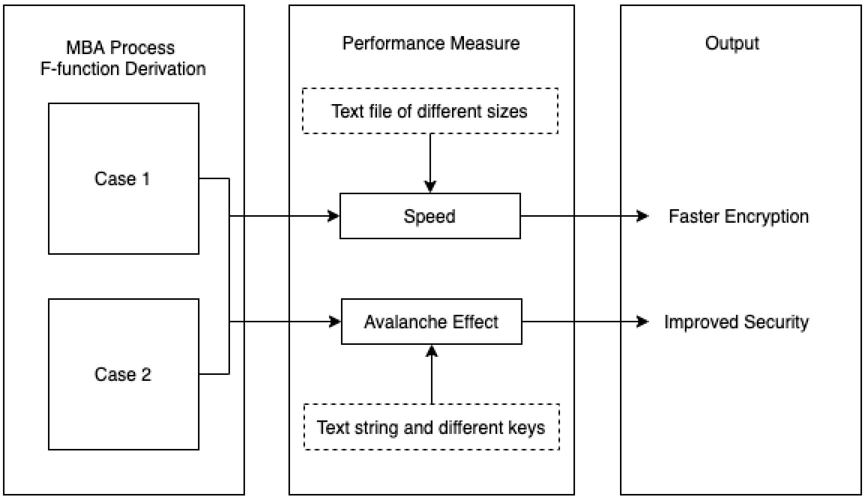


Figure 4. The framework of the MBA analysis using derivation cases

2.4. Development and testing

Diffusion is considered as a desirable property of cryptographic algorithms, reflecting cryptographic strength [16]. It is measured using the avalanche effect. Avalanche effect in this paper ensures that the diffusion property of the modified algorithm was not affected by the removal of the two S-boxes, and that the addition of the derivation process removed the symmetry between the S-boxes.

Avalanche uses hamming distance, a measure of dissimilarity, which is the sum of bit by bit XOR calculation of the equivalent ASCII value. A high avalanche effect is deemed desirable. The formula in getting the avalanche effect is as shown:

Avalanche effect=(hamming distance ÷size)

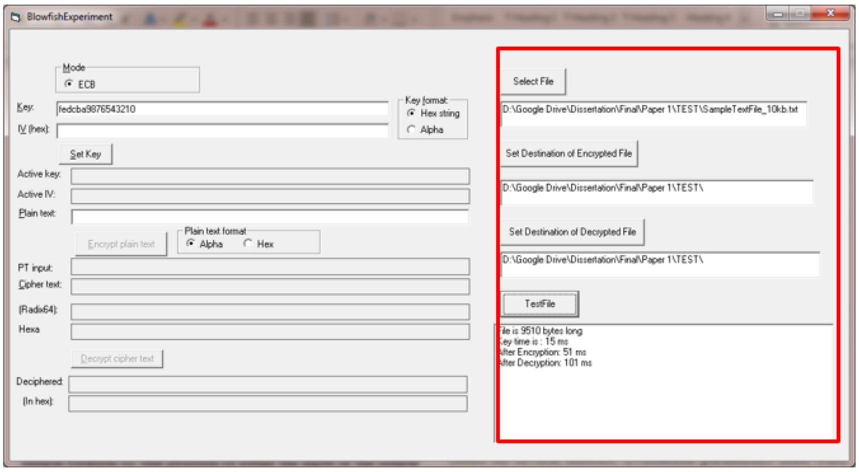
For this test, the hexadecimal values of the encrypted input string with the different keys were used as input in a spreadsheet application to compute for the average avalanche effect. Three plaintext messages were used in the trial, and for one plaintext message, five keys were used, varying 1 bit for each key, as seen in Table 1. An algorithm should possess an avalanche effect minimum of 50% to be considered good [33].

Table 1. Sets of plaintext messages used and keys used

Plaintext Set	Plaintext	Keys
1	“Peter Piper picked a peck of pickled peppers”	0123456789ABCDEF, 0123456789ABCDEE, 0123456789ABCDED, 0123456789ABCDEC, 0123456789ABCDEB.
2	“Leron leron sinta, buko ng papaya”	fedcba9876543210, fedcba9876543211, fedcba9876543212, fedcba9876543213, fedcba9876543214.
3	“Theda Flare Quilala”	a0b0c0d010203040, a0b0c0d010203041, a0b0c0d010203042, a0b0c0d010203043, a0b0c0d010203044.

2.4.1. Performance comparison of MBA using the two derivations

The algorithms were initially downloaded from www.schneier.com. The Visual Basic implementation of David Ireland [34] was adopted for BA. Consequently, modifications were inserted into the original Blowfish algorithm to create the MBA. The selection of files to encrypt and the setting of the encrypted and decrypted file destination were added on Blowfish. The actual timestamp was also added, as can be seen in Figure 5.



Mode: ● ECB

Key: fedcba9876543210

Key format: ● Hex string ○ Alpha

IV (hex):

Active key:

Active IV:

Plain text:

Plain text format: ● Alpha ○ Hex

PT input:

Cipher text:

(Radix64):

Hexa

Deciphered:

(In hex):

D:\Google Drive\Dissertation\Final\Paper 1\TEST\SampleTextFile_10kb.txt

D:\Google Drive\Dissertation\Final\Paper 1\TEST\

D:\Google Drive\Dissertation\Final\Paper 1\TEST\

```
ile is 9510 bytes long
ey time is : 15 ms
lter Encryption: 51 ms
lter Decryption: 101 ms
```

Figure 5. MBA program modification`

After the modification of the algorithm, materials were gathered and prepared for testing. Experimentation was done to test the speed of the algorithms using text files of the following sizes: 10 kB, 20 kB, 50 kB, 100 kB, 200 kB, 500 kB, and 1000 kB. The average time was computed using twenty trials (n=20) of each file size. Testing parameters used (file and key) were the same for all experiments.

Analysis of the performance of the two derivations of the MBA was done based on several metrics. Evaluation parameters used were key generation time, encryption time, and decryption time. Time was measured in milliseconds. The percentage of change was also calculated to compare the amount of change. Note that a positive value indicates a percent increase, and a negative value equates to a percent decrease. The computation is as follows:

$$\text{Percent change} = ((\text{New value Old value})/(\text{Old value}))\times 100\%$$

3. RESULTS AND DISCUSSION

3.1. Avalanche effect improvement of modified Blowfish algorithm derivations

The avalanche effect of the modified Blowfish derivation process one was compared to derivation process two to determine improvement. One plaintext message was used over five keys, varying 1 bit for each key for each test. There were three trial sets in total. Figure 6 shows the avalanche percentage for each test.

In the figure, the first test shows that MBA derivation one had 50.57% avalanche, and the derivation two was at 51.61%. The second test shows that MBA derivation one achieved 49.17% while derivation two acquired 51.41%. On the third test, MBA derivation one attained 47.11%, and derivation two got 51.88%. As the percentage of the avalanche effect gets a higher value, the better will be the security [35], this means that the derivation process two had a better avalanche, thus reflecting better security. The average avalanche effect of the three plaintext messages used with the corresponding keys were shown in Figure 7.

As shown in Figure 7, the second derivation process achieved a 51.63% average avalanche effect, while derivation one achieved 48.95%. The average avalanche effect of MBA derivation two was 51.63%, which surpasses the desired ideal value of 50%. The result clearly showed that derivation two offers better avalanche, as reflected with a 5.47% improvement from derivation one.

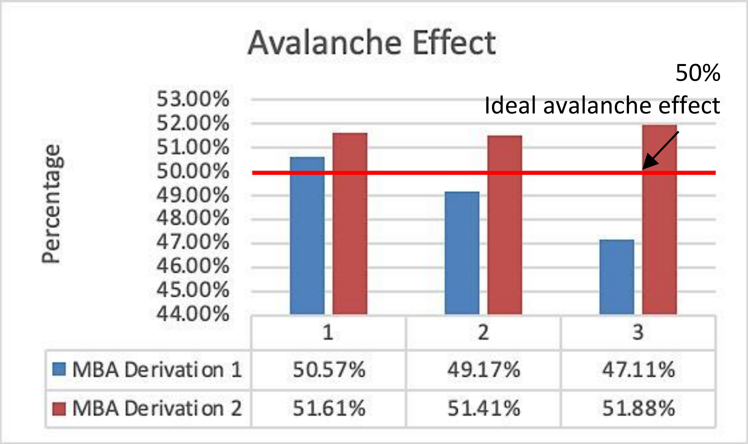


Figure 6. Avalanche effect of three plaintext messages using five keys using the two derivation process

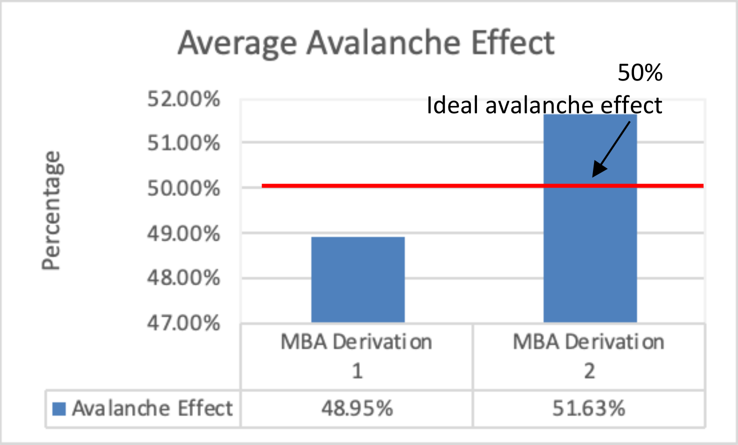


Figure 7. Average avalanche effect of Blowfish and modified Blowfish

3.2. Performance comparison of MBA using the two derivations

The derivation process of the MBA's speed was compared using the execution time of the algorithm's key generation, encryption, and decryption. Experimentation results are shown in Table 2.

Table 2. The key generation time of mba derivation 1 and derivation 2 in milliseconds using different file sizes

Input Size (kB)	Key Generation Time (ms)	
n=20	MBA Derivation 1	MBA Derivation 2
10	23.85	14.50
20	23.85	14.15
50	23.55	14.35
100	23.70	14.40
200	23.90	14.40
500	23.60	14.40
1000	23.90	13.90
Average Key Generation Time (ms)	23.76	14.30

As seen in Table 2, the average key generation time for derivation one was 23.76ms, while derivation two was 14.30ms. This means that the second derivation process is faster by 31.81%. Notice as well that the key generation time is independent of the number of input sizes, which means that. The number of input size does not affect the key generation time. The encryption and decryption time of MBA derivation one and MBA derivation two were shown in Table 3 and Table 4.

Table 3. Encryption time of MBA derivation 1 and derivation 2 in milliseconds using different file sizes

Input Size (kB)	Encryption Time (ms)		Percent
n=20	MBA Derivation 1	MBA Derivation 2	Change (%)
10	81.55	49.75	38.99
20	139.60	84.90	39.18
50	321.50	196.40	38.91
100	625.00	379.80	39.23
200	1210.90	742.20	38.71
500	2984.00	1831.40	38.63
1000	5991.85	3432.20	42.72
	Average Percentage of Change		39.48

Table 4. Decryption time of MBA derivation 1 and derivation 2 in milliseconds using different file sizes

Input Size (kB)	Decryption Time (ms)		Percent Change (%)
n=20	MBA Derivation 1	MBA Derivation 2	
10	142.55	93.15	34.65
20	267.95	163.35	39.04
50	626.45	385.95	38.39
100	1229.55	752.20	38.82
200	2412.20	1471.70	38.99
500	6002.50	3670.70	38.85
1000	12049.15	7274.45	39.63
	Average Percentage of Change		38.34

In the encryption and decryption time presented in Table 3 and Table 4, MBA derivation two consumed less time, thus provides better performance in terms of speed. In the encryption time for the different file sizes, the average percent of change was computed at 39.48%, and in the average decryption time, the change was computed at 38.34%. The result determined that MBA derivation two has faster encryption and decryption time. The difference in time was attributed to the placement of the shift. Results showed that the encryption and decryption time increase as the input file size also increases. The relationship between the time and size is directly proportional to the file size.

4. CONCLUSION

The improvement of MBA derivation case one compared to case two in terms of avalanche effect was determined to be 5.47%. MBA derivation two has better security than the first derivation. The performance of the MBA derivation one over MBA derivation two in terms of time was determined to be: 31.81% slower in the key generation; 39.48% slower in encryption; and 38.34% slower in decryption. The results presented clearly provides proof that the second derivation process made on the Blowfish algorithm to accommodate 128-bit block size and 128-bit key using the original structure of Blowfish was able to provide better performance based on avalanche criteria and speed. For further improvement, hardware optimization implementation of the modified algorithm with derivation two can be done to lessen the time in the key generation, encryption, and decryption. The use of different block cipher mode operation, block size, and other key size considerations can also be done. Researchers may explore other security measures, aside from the avalanche effect, to further analyze the performance of the MBA. Lastly, this modified encryption can be used for encrypting text files, images, and non-text data as an additional supplementary attachment in Electronic Medical Record implementation.

REFERENCES

[1] S. Oukili and S. Bri, “High throughput parallel implementation of Blowfish algorithm,” *Applied Mathematics & Information Sciences*, vol. 10, no. 6, pp. 2087-2092, 2016, doi: 10.18576/amis/100611.

[2] E. R. Arboleda, J. L. Balaba, and J. C. L. Espineli, “Chaotic rivest-shamir-adlerman algorithm with data encryption standard scheduling,” *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 3, pp. 219-227, 2017, doi: 10.11591/eei.v6i3.627.

[3] W. Alexan and F. Hemeida, "Security Through Blowfish and LSB Bit–Cycling With Mathematical Sequences," *2019 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2019, pp. 229-234, doi:

10.23919/SPA.2019.8936812.

[4] R. R. Corpuz, B. D. Gerardo, and R. P. Medina, “A modified approach of Blowfish algorithm based on S-box permutation using shuffle algorithm,” *In Proceedings of the 2018 VII International Conference on Network, Communication and Computing,* 2018, pp. 140-145, doi: 10.1145/3301326.3301331.

[5] R. Rahim, *et al.*, “An application data security with lempel-ziv welch and Blowfish,” *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 71-73, 2018.

[6] H. V. Gamido, M. V. Gamido, and A. M. Sison, “Developing a secured image file management system using modified AES,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1461-1467, 2019, doi: 10.11591/eei.v8i4.1317.

[7] M. Faheem, S. Jamel, A. Hassan, Z. A., N. Shafinaz, and M. Mat, “A Survey on the Cryptographic Encryption Algorithms,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, pp. 333-344, 2017, doi: 10.14569/IJACSA.2017.081141.

[8] M. Bhattacharya, K. Pal, G. Ghosh and S. S. Mandal, "Generation of novel encrypted code using cryptography for multiple level data security for Electronic Patient Record," *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015, pp. 916-921, doi: 10.1109/BIBM.2015.7359806.

[9] A. R. Krishna, A. S. N. Chakravarthy, and A. S. C. S. Sastry, “A Hybrid Cryptographic System for Secured Device to Device Communication,” *International Journal of Electrical and Computer Engineering*, vol. 6, no. 6, p. 2962, Dec. 2016, doi: 10.11591/ijece.v6i6.pp2962-2970.

[10] M. Ranjan, A. H. Mondal, and M. Saikia, “A Cloud Based Secure Voting System using Homomorphic Encryption for Android Platform,” *International Journal of Electrical and Computer Engineering*, vol. 6, no. 6, p. 2994, Dec. 2016, doi: 10.11591/ijece.v6i6.pp2994-3000.

[11] M. A. Sadikin and R. W. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with digital signature for secure electronic health record application," *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016, pp. 387-392, doi: 10.1109/ISITIA.2016.7828691.

[12] Shivaputra, H. Sheshadri, and V. Lokesha, “A Naïve Visual Cryptographic Algorithm for the Transfer of Compressed Medical Images,” *Bulletin of Electrical Engineering and Informatics*, vol. 5, no. 3, pp. 347-365, 2016, doi: 10.11591/eei.v5i3.544.

[13] J. E. Camargo, D. F. Sierra, and Y. F. Torres, “Study of cryptographic algorithms to protect electronic medical records in mobile platforms,” *Indian Journal of Science and Technology*, vol. 8, no. 21, pp. 1-7, 2015, doi: 10.17485/ijst/2015/v8i21/60739.

[14] H. Abdulrahman, N. Poh and J. Burnett, "Privacy preservation, sharing and collection of patient records using cryptographic techniques for cross-clinical secondary analytics," *2014 IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE)*, 2014, pp. 148-153, doi: 10.1109/CICARE.2014.7007847.

[15] S. Fong-In, S. Kiattisin, A. Leelasantitham and W. San-Um, "A partial encryption scheme using absolute-value chaotic map for secure electronic health records," *The 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)*, 2014, pp. 1-5, doi: 10.1109/JICTEE.2014.6804083.

[16] B. Dakhare, N. N. Shinde, S. S. Salvi, A. H. Kadam, and P. G. Wagh, “Performance Analysis of Data Encryption Algorithms using AES BLOWFISH and SNAP,” *International Journal of Engineering Science*, vol. 8, no. 3, pp. 16466-16468, 2018.

[17] M. Vanitha and R. Mangayarkarasi, “Comparative study of different cryptographic algorithms,” *International Journal of Emerging Trends & Technology in Computer Science*, vol. 8, no. 4, pp. 26433–26438, 2016.

[18] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, “A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish,” *Procedia Computer Science*, vol. 78, no. December 2015, pp. 617-624, 2016, doi: 10.1016/j.procs.2016.02.108.

[19] M. Nazeh, A. Wahid, A. Ali, and M. Marwan, “A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention,” *Journal of Computer Science Applications and Information Technology*, pp. 1-7, 2018.

[20] B. Schneier, “Description of a new variable-length key, 64-bit block cipher (Blowfish),” in *Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K., December 9--11,1993 Proceedings*, R. Anderson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 191–204, doi: 10.1007/3-540-58108-1_24.

[21] R. Ahmad, A. A. Manaf, and W. Ismail, “Implementation of a High-Performance Blowfish for Secure Wireless Communication,” *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 8, no. 6, pp. 147-151, 2016.

[22] G. Yadav and A. Majare, “A Comparative Study of Performance Analysis of Various Encryption Algorithms,” *nternational Conference On Emanations in Modern Technology and Engineering*, vol. 5, no. 3, pp. 70-73, 2017.

[23] S. W. Jang, “Comparative Analysis of AES, Blowfish, Twofish and Threefish Encryption Algorithms,” *Journal Of Analysis Of Applied Mathematics*, vol. 10, pp. 5-23, 2017.

[24] National Institute of Standards and Technology, “Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard,” *Federal Register*, vol. 62, no. 177, pp. 48051-48058, Oct. 1997.

[25] J. A. Mahdi, “Design and implementation of proposed BR encryption algorithm,” *IJCCCSE*, vol. 9, no. 1, pp. 1–17, 2009, doi: 10.1007/978-981-15-5558-9_36.

[26] A. R. L. Reyes, E. D. Festijo, and R. P. Medina, “Securing One Time Password (OTP) for Multi- Factor Out-of-Band Authentication through a 128- bit Blowfish Algorithm,” *International Journal of Communication Networks and Information Security*, vol. 10, no. 1, pp. 242–247, 2018.

[27] B. F. Cruz, K. N. Domingo, F. E. De Guzman, J. B. Cotiangco, and C. B. Hilario, “Expanded 128-bit Data

Encryption Standard," *Int. J. Comput. Sci. Mob. Comput.*, vol. 68, no. 8, pp. 133–142, 2017, doi: 10.13140/RG.2.2.36392.72969.

[28] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *International Journal of Security and its Applications*, vol. 9, no. 4, pp. 289–306, 2015, doi: 10.14257/ijsia.2015.9.4.27.

[29] N. J. Oishi, A. Mahamud and Asaduzzaman, "Short paper: enhancing Wi-Fi security using a hybrid algorithm of blowfish and RC6," 2016 International Conference on Networking Systems and Security (NSysS), 2016, pp. 1-5, doi: 10.1109/NSysS.2016.7400706.

[30] A. M. Alabaichi, R. Mahmood, F. Ahmad, and M. S. Mechee, "Randomness Analysis on Blowfish Block Cipher Using ECB and CBC Modes," *Journal of Applied Sciences*, vol. 13, no. 6, pp. 768-789, Jun. 2013, doi: 10.3923/jas.2013.768.789.

[31] T. F. G. Quilala, A. M. Sison, and R. P. Medina, "Modified Blowfish Algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 38–45, 2018, doi: 10.11591/ijeecs.v12.i1.pp38-45.

[32] T. F. G. Quilala, A. M. Sison, and R. P. Medina, "Securing Electronic Medical Records Using Modified Blowfish Algorithm," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 6, no. 3, 2018, doi: 10.11591/ijeei.v6i3.493.

[33] S. S. Hameed, "SMX Algorithm : A Novel Approach to Avalanche Effect on Advanced Encryption Standard AES," in *5th International Conference on "Computing for Sustainable Global Development,"* 2018, pp. 727-732.

[34] "Blowfish Source Code." [Online]. Available: https://www.schneier.com/academic/Blowfish/download.html.

[35] B. S. Ross and V. Josephraj, "Performance Enhancement of Blowfish Encryption Using RK-Blowfish Technique," *International Journal of Applied Engineering Research*, vol. 12, no. 20, pp. 9236–9244, 2017.

BIOGRAPHIES OF AUTHORS

Theda Flare G. Quilala is currently an Associate Professor in the College of Computer Studies at Tarlac State University, Tarlac City, Philippines. A Doctor of Information Technology graduate at Technological Institute of the Philippines and a CHED K-12 scholar. Research interest includes security, data mining, and algorithms.

Rogel Ladia Quilala is an Assistant Professor of ICT at the Tarlac State University-College of Computer Studies. He received his Doctor in Information Technology (DIT) degree from Technological Institute of the Philippines (TIP). He has 18 years of academic experience teaching ICT courses in the tertiary level. During this time, he had a stint as exchange professor in IT at YeungJin College in South Korea.